# Q & A

READERS' QUESTIONS, EDITORS' ANSWERS

## Control Via PC Parallel Port

**Q** *Could you explain how to turn a 12-volt DC relay on for 10 seconds, then off for 20 seconds, and repeat this cycle 1000 times, using a PC programmed in BASIC? The relay coil pulls about 250 mA. Ideally, I would like to be able to control three or four relays from a BASIC program. — L. H. B., Burton, MI*

**Q** *I want to write an automatic control program for a PC and send the input and output signals through the serial port or parallel port. Could you give me complete information about the serial and parallel ports? — Y. S. X., Trenton, NJ*

A PC parallel ports (printer ports) are extremely handy. Each port provides eight data lines plus several control lines; the whole thing is very similar to the bus of an 8-bit microprocessor, so it's easy to interface to digital ICs and other circuitry.

Taking the first question first, you can control up to eight relays with the data lines of a parallel port using the circuit shown in Fig. 1. The relay coils are controlled by IRF510 power switching transistors driven by the data from the PC. You'll need a separate source of 12 volts unless you're sure your PC can spare 250 mA for each relay coil; in that case, you can take 12 volts from a disk-drive power connector.

The program is shown in Listing 1. It retrieves the port address of printer port LPT1 and then transmits data directly to the port, bypassing the operating system in order to ignore printer status signals ("ready," "out of paper," and the like). If you want to use port LPT2, change the PEEK functions so that they look at locations 10 and 11 instead of 8 and 9 respectively.

To count seconds, the program uses a simple trick. The BASIC function TIME$ gives the time of day in hours, minutes, and seconds. The value of TIME$ therefore changes exactly once per second, and when it has changed ten times, ten seconds have elapsed.

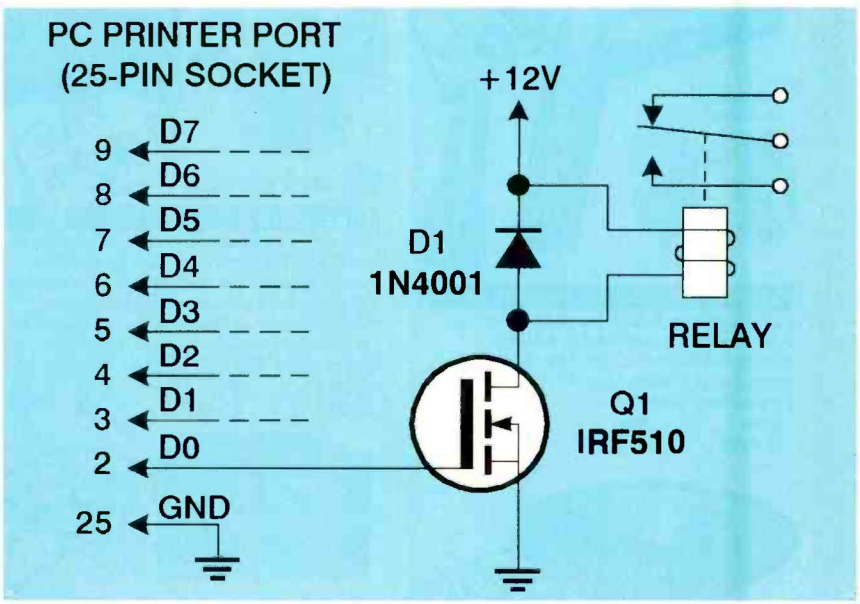


**FIG. 1**—EACH DATA BIT of a PC's parallel port can be used to control a relay using this simple interface circuit.

The sample program controls only one relay connected to data line D0. To control multiple relays independently of each other, use the statement

OUT p, INP(p) OR 2^N

to turn relay number N on, or

OUT p, INP(p) AND NOT 2^N

to turn relay N off, leaving the other relays unchanged. Here N ranges from 0 to 7.

You can also use the parallel port for input, as described in this column in October 1996. The definitive handbook on unconventional uses of parallel ports is *Parallel Port Complete*, by Jan Axelson, available through bookstores and by direct mail from Lakeview Research, Madison, WI 53704; Web: www.lvr.com. This is an easy-to-understand book full of practical circuits as well as valuable reference information. The companion volume, *Serial Port Complete*, will also be available by the time you read this.

**LISTING 1**

```
100 ' Get port address of LPT1
110 DEF SEG = &H40
120 P = PEEK(8) + 256 * PEEK(9)
130 ' Repeat 1000 times...
140 FOR I = 1 TO 1000
150   ' Turn D0 on
160   OUT P, 1
170   ' Wait 10 seconds
180   FOR J = 1 TO 10
190     T$ = TIME$
200     WHILE T$ = TIME$: WEND
210   NEXT J
220   ' Turn D0 off
230   OUT P, 0
240   ' Wait 20 seconds
250   FOR J = 1 TO 20
260     T$ = TIME$
270     WHILE T$ = TIME$: WEND
280   NEXT J
290 ' End of loop
300 NEXT I
```



www.americanradiohistory.com